

PATENT ABSTRACTS OF JAPAN

2

(11)Publication number : 07-084963
 (43)Date of publication of application : 31.03.1995

(51)Int.Cl.

G06F 15/16
 G06F 13/14

(21)Application number : 05-231233
 (22)Date of filing : 17.09.1993

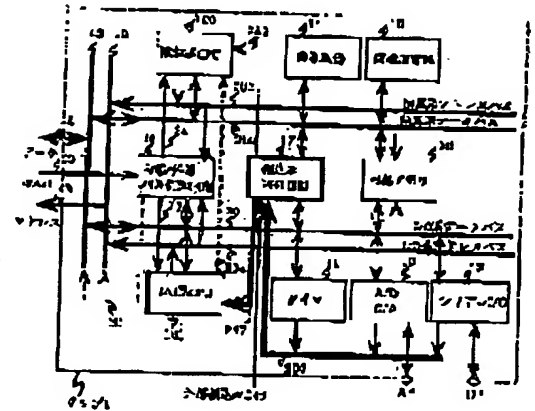
(71)Applicant : HITACHI LTD
 (72)Inventor : OSUGA HIROSHI
 NOGUCHI YOSHIKI
 UCHIYAMA KUNIO
 NENE YOSHITO
 IKEDA HIROSHI
 AMANO AKIO
 KAMIMAKI HARUO
 ASAKAWA YOSHIAKI

(54) SEMICONDUCTOR INTEGRATED CIRCUIT WITH CPU

(57)Abstract:

PURPOSE: To improve the efficiency of the data arithmetic processing of multiple CPUs by performing an interruption request processing from a peripheral module at a high speed.

CONSTITUTION: Peripheral modules 11, 12, and 13 are connected to an I/O system data bus 201, an interruption control circuit 17 is connected to an I/O system processor 101 and the peripheral modules, and an interruption control circuit 17 transfers interruption requests from the peripheral modules to the I/O system processor 101. The peripheral modules need to perform data transfer through an I/O system data bus 102 and selecting circuits 18 and 19 when sending or receiving data to or from outside the chip. The data transfer between the peripheral modules and the outside of the chip through the I/O system data bus 102 is performed under the control of the I/O system processor 101, so an interruption request for this data transfer is not sent to an arithmetic system processor 100.



LEGAL STATUS

Date of request for examination]

Date of sending the examiner's decision of rejection]

Kind of final disposal of application other than the
examiner's decision of rejection or application
converted registration]

Date of final disposal for application]

Patent number]

Date of registration]

Number of appeal against examiner's decision of
rejection]

Date of requesting appeal against examiner's decision

BEST AVAILABLE COPY

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平7-84963

(43)公開日 平成7年(1995)3月31日

(51)IntCl.

G O 6 F 15/16
13/14

識別記号

厅内整理番号

FI

技術表示箇所

S 7429-5L

310 J 8133-5B

審査請求 未請求 請求項の数5 〇L (全 23 頁)

(21) 出願番号

特製平5-231233

(22) 出願日

平成5年(1993)9月17日

(71) 出題人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 究明者 大須賀 宏

東京都国分寺市東荏ヶ谷1丁目280番地

株式会社日立製作所中央研究所内

(72)発明者 野口 季樹

東京都国分寺市東盛ヶ丘1丁目280番地

株式会社日立製作所中央研究所内

(72) 聰明者 内山 邦男

東京都国分寺市東恋ヶ窪1丁目280番地

株式会社日立製作所中央研究所内

(74) 代理人 弁理士 小川 勝男

最終頁に続く

(54) 【発明の名称】 CPUを有する半導体集積回路

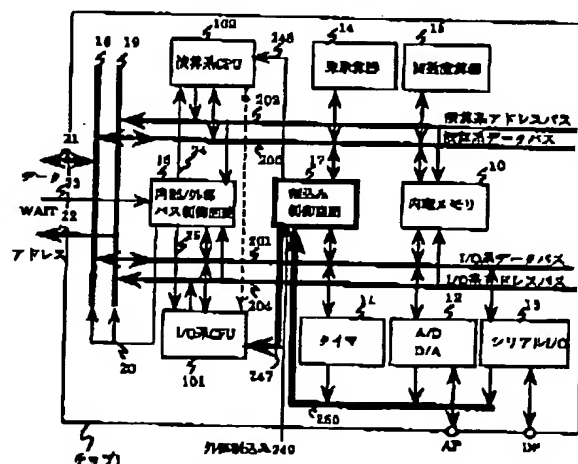
(57) 【要約】

【目的】 マルチCPUにおいて、周辺モジュールからの割込み要求処理を高速に実行し、CPUのデータ演算処理の効率を向上させる。

【構成】 I/O系データバス201には周辺モジュール11、12、13が接続され、I/O系プロセッサ101と周辺モジュールには割込み制御回路17が接続され、割込み制御回路17は周辺モジュールからの割込み要求をI/O系プロセッサ101に転送する。

【効果】 周辺モジュールはチップ外部とのデータの送信もしくは受信に際して、I/O系データバス102と選択回路18,19とを介してのデータ転送を行う必要が有る。I/O系データバス102を介してのチップ外部と周辺モジュールとのデータ転送はI/O系プロセッサ101の管理下で行われるので、このデータ転送のための割込み要求が演算系プロセッサ100に伝達されない。

图 1



1

【特許請求の範囲】

【請求項1】アドレスを出力する第1と第2のプロセッサと、

上記第1のプロセッサと接続された第1のアドレスバスおよび第1のデータバスと、

上記第2のプロセッサと接続された第2のアドレスバスおよび第2のデータバスと、

上記第1と第2のアドレスバスおよび上記第1の第2のデータバスとに接続された内蔵メモリと、

上記第1と第2のアドレスバスの選択された一方と上記第1の第2のデータバスの選択された一方とを外部アドレスバス(22)と外部データバス(21)とに接続する選択回路とを具備してなり、

上記第2のデータバスには、タイマ、A/D変換器およびD/A変換器、シリアル入出力インターフェースの少なくともひとつである周辺モジュールが接続され、

上記第2のプロセッサと上記周辺モジュールには割込み制御回路が接続され、

上記割込み制御回路は上記周辺モジュールからの割込み要求を上記第2のプロセッサに転送することを特徴とする半導体集積回路。

【請求項2】上記割込み制御回路はさらに上記第1のプロセッサと接続され、

上記第2のプロセッサによる外部データの上記内蔵メモリへの転送終了時に生成される上記第2のプロセッサからの割込み要求を上記割込み制御回路は上記第1のプロセッサに転送することを特徴とする請求項1に記載の半導体集積回路。

【請求項3】上記第1と上記第2のプロセッサの一方を指定する指定手段をさらに具備してなり、

上記第1のプロセッサからの複数のアクセスと上記第2のプロセッサからの複数のアクセスに際して、上記選択回路は上記指定された一方からの上記複数のアクセスを上記指定された一方と上記外部アドレスバスとの接続により独占的に連続して実行することを特徴とする請求項1に記載の半導体集積回路。

【請求項4】上記内蔵メモリは複数のバンクから構成され、

上記複数のバンクへのアクセスを制御する情報を格納する格納手段が上記第2のデータバスに接続され、

上記第1のアドレスバスを介して上記第1のプロセッサから上記複数のバンクへのアクセスを上記第1のアドレスバスの上位ビットと上記格納手段の情報とにตอบสนองして制御する第1のアドレスデコーダが上記第1のアドレスバスに接続されてなり、

上記第2のプロセッサによりアクセスされた外部データの複数のバンクの少なくとも一つのバンクへのデータ転送の終了の度に上記第2のプロセッサは上記第2のデータバスを介して上記格納手段の情報を更新し、

上記第2のプロセッサによりアクセスされた外部データ

(2)

特開平7-84963

2

が複数のバンクの少なくとも一つのバンクへのデータ転送の終了の度に生成される上記第2のプロセッサからの割込み要求を、上記割込み制御回路は、上記第1のプロセッサに転送せしめ、

上記第1のプロセッサは上記データ転送終了時の各割込み要求にตอบสนองして同一のアクセスアドレスを上記第1のアドレスバスへ送出することを特徴とする請求項1に記載の半導体集積回路。

【請求項5】上記第1のデータバスには乗除算器、関数演算器、浮動小数点演算プロセッサの少なくともひとつである補助演算モジュールが接続されてなることを特徴とする請求項1から請求項4までのいずれかひとつに記載の半導体集積回路。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明はCPU(中央処理装置)を有する半導体集積回路に関し、特に、同一のチップ上に2つのCPUを搭載したシングルチップマイクロコンピュータのデータ演算処理の処理効率向上を行うために、同一チップ上にデータ演算処理対応のためのCPUとI/O処理対応(周辺モジュール処理対応)のためのCPUとを設けたシングルチップマイクロコンピュータに関する。

【0002】

【従来の技術】シングルチップマイクロコンピュータの従来例として、日立シングルチップマイクロコンピュータHD6475328がある。このHD6475328では、一つのデータバスにCPU、メモリの他に、タイマやシリアルコミュニケーションインタフェースといった周辺モジュールが接続されている。CPUは、メモリから命令を読み込み、読み込んだ命令に従ってデータを入力、演算、結果の格納という処理を実行する。そのほかにCPUは、周辺モジュールからの割込み要求を受付け、割込みを発生した周辺モジュールに対して割込み要求処理を行う。一方、特開昭62-152064号公報には、複数のCPUが独立に内部アドレスデータバスと内蔵メモリとを所有し、これらの内部アドレスデータバスが共通のバス制御装置を介して外部アドレスデータバスと接続されることにより処理能力の向上を図ったシングルチップマイクロコンピュータが開示されている。一方、周知のように計算機システムでは、動作の柔軟性を実現する手段として割込み機能を備えている。特に、外部割込みは、プログラムの実行と独立な原因で生じる割込みであり、電源異常、タイマ割込み、入出力割込み、外部信号などが具体的な要因である。このような割込み要因が検出されて割込みが生じると、プログラムの実行が中断され、そのプログラムの実行状態を規定するプログラム状態語ないしはプログラム状態ベクタが一定の領域に退避され、割込み処理を行うルーチンへ制御が移される。

【0003】

【発明が解決しようとする課題】CPUを1個のみ有する従来のシングルチップマイクロコンピュータにおいては、CPUが本来実行すべきデータ演算処理の他に周辺モジュールからの割込み要求処理を行わなければならない。データ演算処理の効率が低下していた。すなわち、CPUのデータ演算処理の効率向上に関する考慮がなされていなかった。また特開昭62-152064号公報に開示された複数のCPUを有する従来のシングルチップマイクロコンピュータにおいても、複数のCPUが割込み要求処理をどのように実行すれば、データ演算処理の効率の低下を回避できるかについては開示されていない。

【0004】従って、発明の目的とするところは、複数のCPUを有するシングルチップマイクロコンピュータにおいて、周辺モジュールからの割込み要求処理を高速に実行するとともに、CPUのデータ演算処理の効率を向上させることにある。

【0005】

【課題を解決するための手段】上記目的を達成するために、本願で開示される代表的な実施形態の半導体集積回路は、アドレスを出力する第1と第2のプロセッサ(100,101)と、上記第1のプロセッサ(100)と接続された第1のアドレスバス(202)および第1のデータバス(200)と、上記第2のプロセッサ(101)と接続された第2のアドレスバス(204)および第2のデータバス(201)と、上記第1と第2のアドレスバスおよび上記第1の第2のデータバスとに接続された内蔵メモリ(10)と、上記第1と第2のアドレスバスの選択された一方と上記第1の第2のデータバスの選択された一方とを外部アドレスバス(22)と外部データバス(21)とに接続する選択回路(18,19)とを具備してなり、上記第2のデータバス(201)には、タイマ(11)、A/D変換器およびD/A変換器(12)、シリアル入出力インターフェース(13)の少なくともひとつである周辺モジュールが接続され、上記第2のプロセッサ(101)と上記周辺モジュールとは割込み制御回路(17)が接続され、上記割込み制御回路(17)は上記周辺モジュールからの割込み要求(250)を上記第2のプロセッサ(101)に転送することを特徴とする(図1参照)。

【0006】本発明の好適な実施形態による半導体集積回路は、上記割込み制御回路(17)はさらに上記第1のプロセッサ(100)と接続され、上記第2のプロセッサ(101)による外部データの上記内蔵メモリ(10)への転送終了時に生成される上記第2のプロセッサ(101)からの割込み要求を上記割込み制御回路(17)は上記第1のプロセッサ(100)に転送することを特徴とする(図1参照)。

【0007】本発明の他の好適な実施形態による半導体集積回路は、上記第1と上記第2のプロセッサの一方を指定する指定手段(53)をさらに具備してなり、上記第1のプロセッサ(100)からの複数のアクセスと上記第2の

(3)

4

プロセッサ(101)からの複数のアクセスに際して、上記選択回路(18,19)は上記指定された一方からの上記複数のアクセスを上記指定された一方と上記外部アドレスバスとの接続により独占的に連続して実行することを特徴とする(図21および図23参照)。

【0008】本発明のより好適な実施形態による半導体集積回路は、上記内蔵メモリは複数のバンク(115,116,117,118)から構成され、上記複数のバンクへのアクセスを制御する情報を格納する格納手段(119)が上記第2のデータバス(201)に接続され、上記第1のアドレスバス(202)を介しての上記第1のプロセッサ(100)から上記複数のバンクへのアクセスを上記第1のアドレスバス(202)の上位ビットと上記格納手段(119)の情報とにตอบสนองして制御する第1のアドレスデコーダ(113)が上記第1のアドレスバス(202)に接続されてなり、上記第2のプロセッサ(101)によりアクセスされた外部データの複数のバンク(115,116,117,118)の少なくとも一つのバンクへのデータ転送の終了の度に生成される上記第2のプロセッサ(101)からの割込み要求を上記割込み制御回路(17)は上記第1のプロセッサ(100)に転送せしめ、上記第2のプロセッサ(101)によりアクセスされた外部データの複数のバンク(115,116,117,118)の少なくとも一つのバンクへのデータ転送の終了の度に上記第2のプロセッサ(101)は上記第2のデータバス(201)を介して上記格納手段(119)の情報を更新し、上記第1のプロセッサ(100)は上記データ転送終了時の各割込み要求にตอบสนองして同一のアクセスアドレスを上記第1のアドレスバス(202)へ送出することを特徴とする(図15参照)。

【0009】本発明のより具体的な実施形態による半導体集積回路は、上記第1のデータバス(200)には乗除算器(14)、関数演算器(15)、浮動小数点演算プロセッサの少なくともひとつである補助演算モジュールが接続されてなることを特徴とする(図1参照)。

【0010】

【作用】本願で開示される代表的な実施形態の半導体集積回路(図1参照)では、特に、第2のデータバス(201)にはタイマ(11)、A/D変換器およびD/A変換器(12)、シリアル入出力インターフェース(13)の少なくともひとつである周辺モジュールが接続され、第2のプロセッサ(101)とこの周辺モジュールとは割込み制御回路(17)が接続され、この割込み制御回路(17)は周辺モジュールからの割込み要求を第2のプロセッサ(101)に転送するものであるため、下記の如き動作が可能となる。すなわち、タイマ(11)、A/D変換器およびD/A変換器(12)、シリアル入出力インターフェース(13)等の周辺モジュールはチップ外部とのデータの送信もしくは受信に際して、第2のデータバス(102)と選択回路(18,19)とを介してのデータ転送を行う必要が有る。この第2のデータバス(102)を介してのチップ外部と周辺モジュールとのデータ転送は第2のプロセッサ(101)の管理下で行わ

5

れるので、このデータ転送のための割込み要求が第1のプロセッサ(100)に伝達されることはない。従って、この間に第1のプロセッサ(100)は第1のアドレスバス(202)と第1のデータバス(200)とを介して内蔵メモリ(10)にアクセスでき、このアクセス・データに論理演算の処理を行うことができる。その具体的な内容は、下記の通りである。すなわち、タイマ(11)はタイマ定数レジスタにセットされたデータとクロックに同期して更新されるデータとを比較し、一致するとタイマコントロール/ステータスレジスタへフラグを立てる。また、タイマ(11)のタイマ定数レジスタへのデータのセットは第2のプロセッサ(101)によってチップ外部から選択回路(18,19)と第2のデータバス(201)とを介して行われるが、この間に第1のプロセッサ(100)にはタイマ定数レジスタへのデータのセットのための割込みが伝達されないで第1のプロセッサ(100)は第1のアドレスバス(202)と第1のデータバス(200)とを介して内蔵メモリ(10)にアクセスでき、このアクセス・データに論理演算の処理を行うことができる。さらに、タイマ(11)による上記のカウント動作終了時の割込みに基づいて第1のプロセッサ(100)による内蔵メモリ(10)のデータのアクセス、さらにはアクセス・データの論理演算を可能とするために、タイマ(11)からのカウント動作終了時の割込みは内部要求割込み信号(250)として割込み制御回路(17)を介して第1のプロセッサ(100)へ伝達することが望ましい。また、シリアル入出力インターフェース(13)は内部のデータとシリアルポートのデータをシフトレジスタ部でデータの並列-直列変換もしくは直列-並列変換を行って送受信を行うもので、チップ外部からのデータ受信時には直列データから並列データへの変換を行い、また、チップ外部へのデータ送信時には並列データから直列データへの変換を行う。また、受信時の直列-並列変換および送信時の並列-直列変換に際してチップ外部との接続のためのデジタルポート(DP)と第2のデータバス(201)とを間のデータ転送は第2のプロセッサ(101)の管理によって行われるが、この間に第1のプロセッサ(100)にはこのデータ転送のための割込みが伝達されないで第1のプロセッサ(100)は第1のアドレスバス(202)と第1のデータバス(200)とを介して内蔵メモリ(10)にアクセスでき、このアクセス・データに論理演算の処理を行うことができる。また、シリアル入出力インターフェース(13)による受信時の直列-並列変換の完了の割込みに基づいて変換データの第2のプロセッサ(101)による内蔵メモリ(10)への格納および第1のプロセッサ(100)による内蔵メモリ(10)のデータのアクセス、さらにはアクセス・データの論理演算を可能とするために、シリアル入出力インターフェース(13)からの変換動作終了時の割込みは内部要求割込み信号(250)として割込み制御回路(17)を介して第1のプロセッサ(100)と第2のプロセッサ(101)とへ伝達することが望ましい。また、A/D変換器およびD

(4)

特開平7-84963

6

／A変換器(12)は内部のデータとアナログポートとの間でアナログ-デジタル変換(A/D変換)もしくはデジタル-アナログ変換(D/A変換)を行って送受信を行うもので、チップ外部からのアナログ信号受信時にはアナログデータからデジタルデータへの変換(A/D変換)を行い、また、チップ外部へのアナログ信号送信時にはデジタルデータからアナログデータへの変換(D/A変換)を行う。また、受信時および送信時のA/D変換およびD/A変換に際してチップ外部との接続のためのアナログポート(AP)と第2のデータバス(201)とを間のデータ転送は第2のプロセッサ(101)の管理によって行われるが、この間に第1のプロセッサ(100)にはこのデータ転送のための割込みが伝達されないで第1のプロセッサ(100)は第1のアドレスバス(202)と第1のデータバス(200)とを介して内蔵メモリ(10)にアクセスでき、このアクセス・データに論理演算の処理を行うことができる。また、A/D変換器およびD/A変換器(12)による受信時のA/D変換の完了の割込みに基づいて変換データの第2のプロセッサ(101)による内蔵メモリ(10)への格納および第1のプロセッサ(100)による内蔵メモリ(10)のデータのアクセス、さらにはアクセス・データの論理演算を可能とするために、A/D変換器およびD/A変換器(12)からの変換動作終了時の割込みは内部要求割込み信号(250)として割込み制御回路(17)を介して第1のプロセッサ(100)と第2のプロセッサ(101)とへ伝達することが望ましい。

【0011】本発明の好適な実施形態の半導体集積回路では、割込み制御回路(17)はさらに第1のプロセッサ(100)と接続され、第2のプロセッサ(101)による外部データの内部メモリ(10)への転送終了時に生成される第2のプロセッサ(101)からの割込み要求を割込み制御回路(17)は第1のプロセッサ(100)に転送するであるため、内蔵メモリ(10)中の転送データの第1のプロセッサ(100)によるアクセスと論理演算を速やかに開始させることが可能となる。

【0012】本発明の他の好適な実施形態の半導体集積回路では、第1のプロセッサ(100)からの複数のアクセスと第2のプロセッサ(101)からの複数のアクセスに際して、指定手段(53)によって指定された第1と上記第2のプロセッサの一方からの複数のアクセスを選択回路(18,19)は独占的に連続して実行するものである。従って、ユーザが指定された一方として第2のプロセッサ(101)を指定した場合は、周辺モジュール(11,12,13)とチップ外部とのデータ転送を処理する第2のプロセッサ(101)に外部アドレスバス(22)と外部データバス(21)の独占的連続使用権が付与される。この場合は、チップ内部のデータ論理演算を主に処理する第1のプロセッサ(100)よりも周辺モジュール(11,12,13)の機能が重視されている。逆の場合には、周辺モジュール(11,12,13)よりもチップ内部のデータ論理演算を主に処理する第1のプロ

セッサ(100)の機能が重視されることとなる。いずれの場合にも、種々のユーザの志向に適したアクセスモードが指定手段(53)によって指定されることができる。

【0013】本発明のより好適な実施形態の半導体集積回路(図15参照)では、内蔵メモリは複数のバンク(115, 116, 117, 118)から構成され、複数のバンクへのアクセスを制御する情報を格納する格納手段(119)が第2のデータバス(201)に接続され、第1のアдресバス(202)を介して第1のプロセッサ(100)から複数のバンクへのアクセスを第1のアдресバス(202)の上位ビットと格納手段(119)の情報とにตอบสนองして制御する第1のアдресデコーダ(113)が第1のアдресバス(202)に接続されてなり、第2のプロセッサ(101)によりアクセスされた外部データの複数のバンク(115, 116, 117, 118)の少なくとも一つのバンクへのデータ転送の終了の度に第2のプロセッサ(101)は第2のデータバス(201)を介して格納手段(119)の情報を更新し、第2のプロセッサ(101)によりアクセスされた外部データが複数のバンク(115, 116, 117, 118)の少なくとも一つのバンクへのデータ転送の終了の度に生成される第2のプロセッサ(101)からの割込み要求を割込み制御回路(17)は第1のプロセッサ(100)に転送せしめ、第1のプロセッサ(100)はデータ転送終了時の各割込み要求にตอบสนองして同一のアクセスアドレスを第1のアдресバス(202)へ送出するものである。従って、一つのバンクへのデータ転送の終了の度に生成される第2のプロセッサ(101)からの各割込み要求にตอบสนองして第1のプロセッサ(100)が同一のアクセスアドレスを第1のアдресバス(202)へ送出するにもかかわらず、格納手段(119)の更新情報により同一のアクセスアドレスはその都度アドレス変換される。この変換アドレスは複数のバンク(115, 116, 117, 118)を次々に指定するので、第1のプロセッサ(100)からの同一のアクセスアドレスによって複数のバンク(115, 116, 117, 118)のデータが次々にアクセスされることができる。従って、第1のプロセッサ(100)のアドレス空間中での複数のバンク(115, 116, 117, 118)をアクセスするためのアドレス領域を削減でき、その他のアドレス領域をチップ内部のデータ論理演算を主に処理する第1のプロセッサ(100)のためのプライベートのメモリおよびレジスタ等の記憶手段に割り当てることができる。

【0014】本発明のより具体的な実施形態で半導体集積回路(図1参照)は、第1のデータバス(200)には乗除算器(14)、関数演算器(15)、浮動小数点演算プロセッサの少なくともひとつである補助演算モジュールが接続されているので、チップ内部のデータ論理演算を主に処理する第1のプロセッサ(100)の演算機能がこの補助演算モジュールによって補助されることができる。本発明のその他の目的と特徴は、以下の実施例から明らかとなる。

【0015】

【実施例】以下、本発明の実施例を、図面を用いて詳細に説明する。本発明の実施例においては、基本的に、1チップ内に命令セットのほとんどの命令を実行可能な2つのプロセッサ100, 101を内蔵し、それぞれのプロセッサに接続される2組の内部バス202, 200; 204, 201を備えることで、データ演算処理用のプロセッサ100と周辺モジュール処理用のプロセッサ101とに分けることができる。さらに、プロセッサ101と周辺モジュール11, 12, 13とはは割込み制御回路17が接続され、割込み制御回路17は周辺モジュール11, 12, 13からの割込み要求250を周辺モジュール処理用のプロセッサ101に転送することにより、データ演算処理用のプロセッサ100に対して周辺モジュール11, 12, 13からの割込み要求が無くなるため、プロセッサ100のデータ演算処理の効率が向上することが可能となる。さらに、2組の内部バス202, 200; 204, 201に選択回路18, 19を接続し、外部ポートよりチップ外部の外部アドレスバス22と外部データバス21への接続が可能となるようにすることで、少ない面積でシングルチップマイクロコンピュータを実現することが可能となる。

【0016】図1に、マルチCPUシングルチップマイクロコンピュータの1チップの内のブロック構成を示す。チップ内部の演算系内部データバス200および演算系アドレスバス202には、演算系CPU100、乗除算器14、音声・画像処理等で有用な三角関数等の関数演算器15が接続され、さらには浮動小数点演算プロセッサ等の補助演算モジュールが接続されることができる。CPU100は加算、減算、反転等の処理を行う論理演算ユニット(ALU)とビットシフト等のシフト処理を行うパレルシフト等を含むのみであり、乗除算、関数演算、浮動小数点演算等に関するCPU100のデータ演算処理機能は充分とは言えない。この補助演算モジュール14, 15は、これらのデータ演算処理機能を補助する。関数演算器15は例えば、参画関数 $\sin X = Y$ の複数の入力 X_0, X_1, \dots, X_n と、その複数の出力 Y_0, Y_1, \dots, Y_n とを格納したルックアップテーブルで構成されることができ、このルックアップテーブルは複数の入力 X_0, X_1, \dots, X_n を格納キーと検索キーとして、格納キーと検索キーと一致により、出力データを出力する連想メモリにより構成することが望ましい。一方、I/O系内部データバス201およびI/O系アドレスバス204には、I/O系CPU101、タイマ11、A/DおよびD/A変換器12、シリアル入出力インターフェース13等の周辺モジュールが接続される。この周辺モジュールの機能と動作は既に説明した通りである。さらに図示されていないが、このI/O系内部データバス201、I/O系アドレスバス204には、従来のシングルチップマイクロコンピュータに組込まれていたDMA(ダイレクトメモリアccess)コントローラ等の内蔵周辺

回路を接続することが望ましい。内蔵メモリ10、内部／外部バス制御回路16、割込み制御回路17は、演算系バス200および202、I/O系内部データバス201およびI/O系アドレスバス204の両者に接続される。また、内蔵メモリ10は、演算系CPU100とI/O系CPU101とからアクセスされるシェアメモリ(共有メモリ)としての機能と、演算系CPU100とI/O系CPU101といずれか一方からアクセスされるプライベートメモリ(私有メモリ)としての機能とを有する。尚、内蔵メモリ10は、スタティックRAM、マスクROM、不揮発性メモリ(EPROM、EEPROM等)等により構成されることができる。演算系CPU100とI/O系CPU101とは、シングルチップマイクロコンピュータの命令セットのほとんどの同一命令コードで動作するCPUである。演算系CPU100は主に音声画像処理等のデータ演算処理を行ない、I/O系CPU101は外部デバイスとこのシングルチップマイクロコンピュータとの間のデータ入出力処理を主に行なう。シングルチップマイクロコンピュータ等、一般のロジックLSIでは、外部ピン数の制約がある。そのため、チップ内部では演算系とI/O系に分離されていたアドレスとデータの内部バス202、200;204、204も、チップ外部とのインタフェース(外部ピン)部分ではマルチプレックスさせる必要が有る。従って、演算系データバス200とI/O系データバス201は、選択回路としてのマルチプレクサ18に入力され、このマルチプレクサ18を介して外部データバス21に接続される。同様に、演算系アドレスバス202とI/O系アドレスバス204は、マルチプレクサ19に入力され、このマルチプレクサ19を介して外部アドレスバス22に接続される。チップ外部のデバイスとのデータの受け渡しは、この外部データバス21、外部アドレスバス22と、バスサイクルの期間延長制御を行なうWAIT制御信号23を用いて行なう。従って、演算系CPU100とI/O系CPU101とは、内部バス202、200;204、204とマルチプレクサ18、19と外部データバス21と外部アドレスバス22とを介して、外部メモリ等の外部デバイスをアクセスすることができる。内部／外部バス制御回路16は、演算系CPU100あるいはI/O系CPU101から出力されるアドレス情報202、204に従い、アドレスマップされたチップ内部の内蔵メモリ10、周辺モジュール11、12、13、補助演算モジュール14、15の各種レジスタ、および、チップ外部の外部デバイスとの間のデータ転送サイクルを制御する。外部デバイスとのデータ転送を行なう場合には、演算系あるいはI/O系のどちらの内部バスと外部デバイスとを接続させるかを制御信号20で指定する。バスサイクル終了時点では、演算系CPU100に対しては制御信号24で、I/O系CPU101に対しては制御信号25で、バスサイクルが

完了したことを知らせ、次のデータ転送要求を行なうように指令する。割込み制御回路17には、タイマ11、A/DおよびD/A変換器12、シリアル入出力インターフェース13等の内蔵の周辺モジュールから出力される内部割込み要求信号250と、チップ外部から入力される外部割込み要求信号249とを入力される。これらの割込み要求をもとに、演算系CPU100に対する割込みは割込み要求信号245で、I/O系CPU101に対する割込みは割込み要求信号247で、割込み処理を起動させる。外部割込み要求信号249は電源異常などによる割込みであり、タイマ11、A/DおよびD/A変換器12、シリアル入出力インターフェース13等の内蔵の周辺モジュールから出力される内部割込み要求信号250については、実施例の前に既に詳細に説明した通りである。尚、割込み処理後に実行すべきプログラムアドレスを指定するためのベクタ番号は、割込み制御回路17と2つのCPU100、101間の専用バス(図1の破線)を介して指定する方法と、図1に示すように2つの内部データバス200、201とを介して指定する方法とを採ることができる。以上説明したように本実施例のマルチCPUシングルチップマイクロコンピュータは、内部/外部バス制御回路16、割込み制御回路17、内蔵メモリ10を演算系CPU100とI/O系CPU101で共有させ、2つのシングルチップマイクロコンピュータを1チップ化させた構成となっている。そのため、共有化させた部分の動作を考えなければ、それぞれのCPU100、101は、従来のシングルチップマイクロコンピュータと同様な動作を行なう。

【0017】図1に示した内蔵メモリ10を具体的に実
現する例として、図2にマルチCPUのシェアドメモリ
部におけるメモリ割当ての構成を示す。図2は、演算系
CPU100のデータ演算処理効率の低下を防ぎつつ、
演算系CPU100およびI/O系CPU101へのシ
ェアドメモリの割り振りを可変にする構成を示した図で
ある。

【0018】図2は、演算系CPU100、アドレスデコーダ102、メモリ割当てレジスタ108、選択回路107、シェアドメモリA104、シェアドメモリB105、I/O系CPU101、アドレスデコーダ103より構成されている。メモリ割当てレジスタ108は、I/O系CPU101から書き込まれた値を2つのアドレスデコーダ102、103と選択回路107に出力する。アドレスデコーダ102および103はメモリ割当てレジスタ108の値により、メモリを選択するCS4信号211、CS5信号212、CS6信号213、CS7信号214を制御する。選択回路107は、メモリ割当てレジスタ108の値により、シェアドメモリA104およびシェアドメモリB105を演算系CPU100とI/O系CPU101のどちらに論理的に接続するかを決めるSL0信号215、SL1信号216を制御

11

する。演算系内部データバス200は、シェアドメモリA104およびシェアドメモリB105に接続されている。演算系アドレスバス202は、演算系下位アドレス220を通してシェアドメモリA104に接続されている。演算系アドレスバス202は、演算系下位アドレス221を通してシェアドメモリB105に接続されている。また演算系アドレスバス202の上位アドレスは、演算系上位アドレス203を通してアドレスデコーダ102に接続されている。I/O系内部データバス201は、シェアドメモリA104およびシェアドメモリB105に接続されている。I/O系アドレスバス204は、I/O系下位アドレス222を通してシェアドメモリA104に接続されている。I/O系アドレスバス204は、I/O系下位アドレス223を通してシェアドメモリB105に接続されている。またI/O系アドレスバス204の上位アドレスは、I/O系上位アドレス205を通してアドレスデコーダ103に接続されている。メモリ割当てレジスタ108は、信号線206を通してI/O系内部データバス201に接続されている。メモリ割当てレジスタ108は、信号線209、210を通してアドレスデコーダ102および選択回路107に接続されている。メモリ割当てレジスタ108は、信号線207、208を通してアドレスデコーダ103に接続されている。アドレスデコーダ102より出力されたCS4信号211およびCS5信号212は、それぞれシェアドメモリA104およびシェアドメモリB105に接続されている。アドレスデコーダ103より出力されたCS6信号213およびCS7信号214は、それぞれシェアドメモリA104およびシェアドメモリB105に接続されている。選択回路107より出力されたSL0信号215およびSL1信号216は、シェアドメモリA104およびシェアドメモリB105にそれぞれ接続されている。

【0019】演算系CPU100のメモリマップを図3に示す。この図3より、H' 00000000からH' 7FFFFFFFに外部メモリ空間を、H' 80000000からH' 9FFFFFFFに内部メモリ空間を、H' A0000000からH' BFFFFFFFにメモリ割当て空間0、1をそれぞれ設けており、それ以外にはメモリが設けられていないことがわかる。I/O系CPU101のメモリマップを図4に示す。この図4より、H' 00000000からH' 7FFFFFFFに外部メモリ空間を、H' 80000000からH' 9FFFFFFFに内部メモリ空間を、H' A0000000からH' BFFFFFFFにメモリ割当て空間0、1を、H' C0000000からH' FFFFFFFFに内部メモリ空間をそれぞれ設けていることがわかる。

尚、図2のメモリ割当てレジスタ108にセットされた値に従って、演算系CPU100とI/O系CPU101のメモリ割当て空間0、1にシェアドメモリA104

(7)

特開平7-84963

12

およびシェアドメモリB105を割当てる。

【0020】メモリ割当てレジスタ108の構成を図5に示す。図2に示すように、メモリ割当てレジスタ108は、信号線206およびI/O系内部データバス201を通して、I/O系CPU101より読書き可能である。メモリ割当てレジスタ108は、MD1、MD0の2ビットより構成されている。メモリ割当てレジスタ108の2ビットMD1、MD0値をセットした時の演算系CPU100とI/O系CPU101のメモリマップを図6に示す。図6より、モード00では演算系CPU100側にメモリが全て割当てられ、モード01では演算系CPU100とI/O系CPU101にそれぞれメモリが割当てられ、モード10では演算系CPU100側にはメモリが割当てられないことがわかる。モード11は定義していない。

【0021】演算系CPU100側のアドレスデコーダ102のブロック図を図7に示す。この図より、演算系上位アドレス203のうちA31、A30、A29、A28とMD1、MD0で、CS4信号211とCS5信号212を制御していることがわかる。CS4信号211とCS5信号212は、それぞれシェアドメモリA104、シェアドメモリB105を選択するための信号線である。

【0022】図7に示したアドレスデコーダ102の機能を、図8に示す。

【0023】A31=0の時は、外部空間を指定するので、CS4信号211、CS5信号212にはメモリを選択しないために0を出力する。(A31、A30)=(1、1)は、H' C0000000からH' FFFF FFFFの空間を指定し、(A31、A30、A29)=(1、0、0)の時は内部メモリ空間を指定するので、CS4信号211、CS5信号212には0を出力する。(A31、A30、A29、A28)=(1、0、1、0)の時は、メモリ割当て空間0を指定する。この時は、MD1、MD0の値によってCS4信号211、CS5信号212の値が決定される。(MD1、MD0)=(0、0)と(0、1)の時は、メモリ割当て空間0にシェアドメモリA104を割当てるので、シェアドメモリA104を選択するCS4信号211に1、シェアドメモリB105を選択するCS5信号212に0を出力する。(MD1、MD0)=(1、0)の時は、メモリ割当て空間0にはメモリを割当てないので、CS4信号211、CS5信号212ともに0を出力する。(MD1、MD0)=(1、1)の時は、モードを定義していないので、CS4信号211、CS5信号212ともにdon't care(以後 *と記述)とする。(A31、A30、A29、A28)=(1、0、1、1)の時は、メモリ割当て空間1を指定する。この時は、MD1、MD0の値によってCS4信号211、CS5信号212の値が決定される。(MD1、MD0)=(0、

13

0)の時は、メモリ割当て空間1にシェアドメモリB105を割当てるので、シェアドメモリA104を選択するCS4信号211に0、シェアドメモリB105を選択するCS5信号212に1を出力する。(MD1, MD0) = (0, 1)と(1, 0)の時は、メモリ割当て空間1にはメモリを割当てないので、CS4信号211、CS5信号212ともに0を出力する。(MD1, MD0) = (1, 1)の時は、モードを定義していないので、CS4信号211、CS5信号212ともに*とする。

【0024】図2のI/O系CPU101側のアドレスデコーダ103のブロック図を図9に示す。この図より、I/O系上位アドレス205のうちA31、A30、A29、A28とMD1、MD0で、CS6信号213とCS7信号214を制御していることがわかる。

【0025】図9に示したアドレスデコーダ103の機能を、図10に示す。図10の機能表の考え方は、図8の機能表の考え方と基本的には同じなので詳細な説明は省略する。図8と異なるところは、モードを指定した時にメモリ割当て空間へのメモリの割当て方が演算系CPU100とI/O系CPU101では正反対のため、CS6信号213、CS7信号214の出力値が、CS4信号211、CS5信号212の値と異なるという点である。

【0026】図11に選択回路107のブロック図を示す。この図より、MD1、MD0でSL0信号215、SL1信号216を制御していることがわかる。

【0027】選択回路107の機能を、図12に示す。機能表において、SL信号は、シェアドメモリに入力されている演算系とI/O系のCS信号、下位アドレスおよび内部データバスの選択信号であり、信号が1の時は演算系のCS信号、下位アドレスおよび内部データバスを選択し、信号が0の時はI/O系のものを選択する。

(MD1, MD0) = (0, 0)の時は、シェアドメモリA104およびシェアドメモリB105を演算系CPU100に接続するので、SL0信号215、SL1信号216ともに1を出力する。(MD1, MD0) = (0, 1)の時は、シェアドメモリA104のみを演算系CPU100に接続するので、SL0信号215に1、SL1信号216に0を出力する。(MD1, MD0) = (1, 0)の時は、演算系CPU100にはメモリを接続しないので、SL0信号215、SL1信号216ともに0を出力する。(MD1, MD0) = (1, 1)の時はサポートしていないモードなので、SL0信号215、SL1信号216ともに*とする。

【0028】図13に図2のシェアドメモリA104の構成図を示す。このシェアドメモリA104は、メモリ109、選択回路110、111、112より構成されている。選択回路110は、SL0信号215の値によって演算系下位アドレス220とI/O系下位アドレス

(8)

特開平7-84963

14

バス222の選択を行い、信号線217に出力する。選択回路111は、SL0信号215の値によって演算系内部データバス200とI/O系内部データバス201の選択を行い、信号線218に出力する。選択回路112は、SL0信号215の値によって、CS4信号211とCS6信号213の選択を行い、信号線219に出力する。選択回路110、111および112は、SL0信号が1の時にCS4信号211、演算系下位アドレス220および演算系内部データバス200を選択し、SL0信号215が0の時にCS6信号213、I/O系下位アドレス222およびI/O系内部データバス201を選択する回路である。シェアドメモリB105も図13の構成と同じである。以下動作について述べる。今、モード00を指定してシェアドメモリA104とシェアドメモリB105を2つとも演算系CPU100側に接続する場合を考える。この時、メモリ割当てレジスタ108に00を設定する。始めに演算系CPU100がメモリ割当て空間0へアクセスした場合について述べる。この時(A31, A30, A29, A28) = (1, 0, 1, 0)となる。メモリ割当てレジスタ108は、(MD1, MD0) = (0, 0)となっているので、図8の機能表よりシェアドメモリA104を選択するCS4信号211のみに1が出力される。また、選択回路107は、(MD1, MD0) = (0, 0)となっているので、図12の機能表より、演算系CPU100側にシェアドメモリA104およびシェアドメモリB105を接続するためにSL0信号215、SL1信号216に1が出力される。シェアドメモリA104は、SL0信号215が1となっているため、演算系CPU100からのCS4信号211、演算系下位アドレス220、演算系内部データバス200に接続される。よって演算系CPU100からシェアドメモリA104へのアクセスが可能となる。シェアドメモリB105も演算系CPU100側に接続されるが、CS5信号212が0のため、演算系CPU100からはアクセスされない。次に演算系CPU100が、メモリ割当て空間1へアクセスした場合を考える。図8の機能表より(A31, A30, A29, A28) = (1, 0, 1, 1)、(MD1, MD0) = (0, 0)なので、CS5信号212のみ1が出力される。選択回路107からは、演算系CPU100側にシェアドメモリA104およびシェアドメモリB105が接続されるように、SL0信号215、SL1信号216ともに1が出力される。シェアドメモリB105は、SL1信号216が1となっているため、演算系CPU100からのCS5信号212、演算系下位アドレス222、演算系内部データバス200に接続される。よって演算系CPU100からシェアドメモリA104も演算系CPU100側に接続されるが、CS4信号211が0のため、演算系CPU100からは

15

アクセスされない。一方、I/O系CPU101がメモリ割当て空間0およびメモリ割当て空間1にアクセスすることを考える。図10の機能表より、CS6信号213およびCS7信号214には1が出力されない。また、選択回路107から出力されるSL0信号215、SL1信号216にはシェアドメモリA104とシェアドメモリB105を演算系CPU100に接続させるために1が出力されている。そのため、シェアドメモリA104およびシェアドメモリB105は演算系CPU100に接続されて、I/O系CPU101には接続されない。従って、I/O系CPU101からは、アクセスできない。以上のことより、モード00を指定した場合は、シェアドメモリA104およびシェアドメモリB105を演算系CPU100側に接続してしまうので、I/O系CPU101からはメモリ割当て空間0およびメモリ割当て空間1にアクセスすることはできなくなることがわかる。モードを01、10に設定しても、上記に述べた動作と同じ動作により図6に示したようなメモリの割当てが行われる。本実施例によれば、メモリ割当てレジスタ108、メモリ割当てレジスタ108で制御可能なアドレスデコーダ102、103および選択回路107を設けることにより、演算系CPU100とI/O系CPU101に割当てするメモリを可変にすることが可能となる。さらにメモリ割当てレジスタ108への読み書きは、I/O系CPU101がすべて行うので、演算系CPU100のデータ演算処理の効率低下を防ぐことができる。さらにI/O系CPU101がメモリ割当てレジスタ108への値の書き込みができるので、チップ動作中にメモリ割当てを定めることが可能である。

【0029】また、チップ動作開始時に外部からメモリ割当てレジスタ108にデータを書き込むことで、ユーザごとにプログラマブルにメモリ割当てを行うこともできる。その構成例を図14に示す。図14はパッド129、パッド130、バストランジスタ131、バストランジスタ132およびメモリ割当てレジスタ108より構成されている。接続関係を以下に示す。パッド129は、信号線260を通してバストランジスタ131に接続されている。書き込み信号線259は、バストランジスタ131に接続されている。バストランジスタ131は、信号線261を通してメモリ割当てレジスタ108のMD1ビットに接続されている。パッド130は、信号線262を通してバストランジスタ132に接続されている。書き込み信号線259は、バストランジスタ132に接続されている。バストランジスタ132は、信号線263を通してメモリ割当てレジスタ108のMD0ビットに接続されている。動作を説明する。チップ動作開始時に、パッド129およびパッド130にメモリ割当てレジスタ108にセットしたい値を印加する。書き込み信号線259に信号を送る。書き込み信号線259の信号は、バストランジスタ131およびバストランジスタ

(9)

特開平7-84963

16

132のゲートに入力される。バストランジスタ131およびバストランジスタ132は、書き込み信号線259より信号が入力されると、ON状態になる。バストランジスタ131がON状態になると、パッド129に印加された値は、信号線260、バストランジスタ131、信号線261を通してメモリ割当てレジスタ108のMD1ビットに入力される。パッド130に印加された値も同様に、信号線262、バストランジスタ132、信号線263を通してメモリ割当てレジスタ108のMD0ビットに入力される。以上のように、メモリ割当てレジスタ108にパッド129、パッド130、バストランジスタ131、バストランジスタ132を接続することにより、チップ動作開始時にユーザがプログラマブルに演算系CPU100とI/O系CPU101へのメモリの割当てが可能になる。上記説明では、信号線259は書き込み信号線としたが、リセット信号線入力しても同様の機能を実現できる。また、メモリ割当てレジスタ108は、不揮発性メモリ(EPROM、EEPROM等)やヒューズで実現することもできる。尚、本実施例では2つのシェアドメモリを用いて説明を行ったが、シェアドメモリの数に対応するだけメモリ割当てレジスタのビット数を増加させれば、2つ以上のシェアドメモリについても同様な効果を実現することが可能である。

【0030】図1に示した内蔵メモリ10を実現する他の実施例として、図15にマルチCPUのシェアドメモリ部におけるバンク切換えの構成を示す。図15は、演算系CPU100のメモリマップ上に存在する内部メモリ空間に4つのシェアドメモリのうち、2つの任意のシェアドメモリを割当てることを可能にする構成を示した図である。この構成を用いれば、演算系CPU100はバンク切換え操作を行うことなく、I/O系CPU101より複数のシェアドメモリに書き込まれたデータを次々に取り込むことが可能となる。すなわち、実施例の前に詳細に説明したように、一つのバンクへのデータ転送の終了の度に生成されるI/O系CPU101からの各割込み要求に応答して演算系CPU100が同一のアクセスアドレスをアドレスバス202へ送出するにもかかわらず、レジスタ119の更新情報により同一のアクセスアドレスはその都度アドレス変換され、この変換アドレスは複数のバンク115、116、117、118を次々に指定するので、演算系CPU100からの同一のアクセスアドレスによって複数のバンク115、116、117、118のデータが次々にアクセスされることができる。図15は、演算系CPU100、I/O系CPU101、アドレスデコーダ113、アドレスデコーダ114、シェアドメモリC115、シェアドメモリD116、シェアドメモリE117、シェアドメモリF118、バンク切換えレジスタ119、割込み制御回路17より構成されている点は図1の実施例と基本的に同一であり、バンク切換えレジスタ119が付加されてい

(10)

特開平7-84963

17

る。すなわち、バンク切換えレジスタ119は、信号線240を通してI/O系内部データバス201に接続され、SA1信号241、SA0信号242、SB1信号243、SB0信号244を通してアドレスデコーダ113に接続されている。アドレスデコーダ113より出力されたCS0信号224、CS1信号225、CS2信号226およびCS3信号227は、それぞれシェアドメモリC115、シェアドメモリD116、シェアドメモリE117およびシェアドメモリF118にそれぞれ接続されている。アドレスデコーダ114より出力されたCS8信号228、CS9信号229、CS10信号230およびCS11信号231は、それぞれシェアドメモリC115、シェアドメモリD116、シェアドメモリE117およびシェアドメモリF118にそれぞれ接続されている。演算系CPU100のメモリマップを図18に示す。この図18より、H' 80000000からH' AFFFFFFFに内部メモリ空間A、Bを設けていることがわかる。I/O系CPU101がバンク切換えレジスタ119に値をセットすることにより、内部メモリ空間A、Bに、シェアドメモリC115、シェアドメモリD116、シェアドメモリE117、シェアドメモリF118の中の任意の2つを割当てて、従って、演算系CPU100は、4つのシェアドメモリを同時に認識することができない。

【0031】I/O系CPU101のメモリマップを図17に示す。この図17は、I/O系CPU101からは、4つのシェアドメモリを同時に認識することができることを示している。よってI/O系CPU101は、シェアドメモリC115、シェアドメモリD116、シェアドメモリE117およびシェアドメモリF118をアクセスすることが可能である。

【0032】バンク切換えレジスタ119の構成を図18に示す。バンク切換えレジスタ119は、図15に示すように信号線240およびI/O系内部データバス201を通して、I/O系CPU101より読み書き可能である。バンク切換えレジスタ119は、SA1、SA0、SB1、SB0の4ビットより構成されている。SA1、SA0の2ビットにより、演算系CPU100のメモリマップ上の内部メモリ空間Aに割当ててシェアドメモリの設定を行う。SB1、SB0の2ビットにより、演算系CPU100のメモリマップ上の内部メモリ空間Bに割当ててシェアドメモリの設定を行う。

【0033】演算系CPU100側のアドレスデコーダ113のブロック図を図19に示す。この図19より、演算系上位アドレス203のうちA31、A28とSA1信号241、SA0信号242、SB1信号243、SB0信号244で、CS0信号224、CS1信号225、CS2信号226およびCS3信号227を制御していることがわかる。CS0信号224、CS1信号225、CS2信号226およびCS3信号227は、

18

それぞれシェアドメモリC115、シェアドメモリD116、シェアドメモリE117、シェアドメモリF118を選択するための信号線である。

【0034】アドレスデコーダ113の機能を、図20に示す。A31=0のときは、外部空間を指定するので、CS0信号224、CS1信号225、CS2信号226およびCS3信号227にはメモリを選択しないために0を出力する。(A31、A28)=(1、0)は、メモリ空間Aを指定する。この場合は、空間A指定のSA1信号241、SA0信号242の値が有効になる。(SA1、SA0)=(0、0)のときは、シェアドメモリC115を指定するCS0信号224のみ1が出力される。(SA1、SA0)=(0、1)のときは、シェアドメモリD116を指定するCS1信号225のみ1が出力される。(SA1、SA0)=(1、0)のときは、シェアドメモリE117を指定するCS2信号226のみ1が出力される。(SA1、SA0)=(1、1)のときは、シェアドメモリF118を指定するCS3信号227のみ1が出力される。(A31、A28)=(1、1)は、メモリ空間Bを指定する。この場合は、空間B指定のSB1信号243、SB0信号244の値が有効になる。(SB1、SB0)=(0、0)のときは、シェアドメモリC115を指定するCS0信号224のみ1が出力される。(SB1、SB0)=(0、1)のときは、シェアドメモリD116を指定するCS1信号225のみ1が出力される。(SB1、SB0)=(1、0)のときは、シェアドメモリE117を指定するCS2信号226のみ1が出力される。(SB1、SB0)=(1、1)のときは、シェアドメモリF118を指定するCS3信号227のみ1が出力される。

以下、動作説明を行う。I/O系CPU101がシェアドメモリC115とシェアドメモリD116に書込んだデータを演算系CPU100に引き渡す動作について説明する。I/O系CPU101は、H' C0000000からH' CFFFFFFFの空間をアクセスしてシェアドメモリC115にデータを書込む。シェアドメモリC115へのデータ書込みが終わったら、次にI/O系CPU101は、H' D0000000からH' DFFFFFFFの空間をアクセスしてシェアドメモリD116にデータを書込む。シェアドメモリC115とシェアドメモリD116にデータを書込み終わったところで、I/O系CPU101はバンク切換えレジスタ119に値をセットする。シェアドメモリC115、シェアドメモリD116、シェアドメモリE117、シェアドメモリF118を表す値をそれぞれ00、01、10、11とすると、演算系CPU100にシェアドメモリC115およびシェアドメモリD116を接続するために、バンク切換えレジスタ119に(SA1、SA0)=(0、0)、(SB1、SB0)=(0、1)の値をセットする。バンク切換えレジスタ119への値の

セットが終了したところで、I/O系CPU101は演算系CPU100へ割込み手段により、シェアメモリC115、シェアメモリD116へのデータセット完了を知らせる。その後I/O系CPU101は、演算系CPU100がアクセスしないシェアメモリE117およびシェアメモリF118へデータの書き込みを行う。一方割込みを受けた演算系CPU100は、それまで実行していたデータ演算処理を中断し、内部メモリ空間Aおよび内部メモリ空間Bへのデータセットが完了したと認識して、内部メモリ空間Aおよび内部メモリ空間Bへアクセスを開始する。内部メモリ空間Aへアクセスするときは、(A31, A28) = (1, 0)となる。バンク切換えレジスタ119のSA1信号241、SA0信号242には(SA1, SA0) = (0, 0)がセットされているので、シェアメモリC115を選択するCS0信号224のみに1が出力される。従って、内部メモリ空間Aへアクセスすると、シェアメモリC115をアクセスできる。内部メモリ空間Bへアクセスするときは、(A31, A28) = (1, 1)となる。バンク切換えレジスタ119のSB1信号243、SB0信号244には(SB1, SB0) = (0, 1)がセットされているので、シェアメモリD116を選択するCS1信号225のみに1が出力される。従って、内部メモリ空間Bへアクセスすると、シェアメモリD116をアクセスできる。シェアメモリC115およびシェアメモリD116へのアクセスが終了すると、演算系CPU100はI/O系CPU101へ割込みを発生させ、アクセスが終了したことを知らせる。その後、演算系CPU100は本来のデータ演算処理に復帰する。一方I/O系CPU101は、シェアメモリE117およびシェアメモリF118へのデータセットが完了していたら、バンク切換えレジスタ119に値をセットして、再び演算系CPU100に割込みを発生させる。以上のことより、I/O系CPU101がメモリへの値のセットおよびバンク切換え操作を行うので、演算系CPUのデータ演算処理効率の低下を防ぎつつ、バンク切換えが可能となる。尚、演算系CPU100の内部メモリ空間AにあるシェアメモリC115、E117は演算系CPU100の同一のアクセスアドレスH' 80000000からアクセスされることができ、演算系CPU100の内部メモリ空間BにあるシェアメモリD116、F118は演算系CPU100の同一のアクセスアドレスH' 90000000からアクセスされることが

【0035】図21は、図1の内部/外部バス制御回路16の詳細を示している。内部/外部バス制御回路16の内部は、外部バス制御回路50、演算系内部バス制御回路51、I/O系内部バス制御回路52の3つの順序回路と、アクセス権設定レジスタ53、および2つのOR論理ゲート54、55から構成されている。外部バス

制御回路50には、演算系アドレスバス202、I/O系アドレスバス204、アクセス権設定レジスタ53およびチップ外部から入力されるバスサイクルの期間延長制御を行なうWAIT制御信号23のデータが入力される。これらの入力データをもとに、外部バス制御回路50は、順序回路の内部状態を遷移させるとともに、演算系データバス200とI/O系内部データバス201のマルチプレクサ18および演算系アドレスバス202とI/O系アドレスバス204のマルチプレクサ19の制御信号20、演算系CPU100の外部アクセス終了信号56、I/O系CPU101の外部アクセス終了信号58を出力する。アクセス権設定レジスタ53は、I/O系内部データバス201に接続され、I/O系CPU101がデータを読み書きできる構成になっている。演算系内部バス制御回路51には、演算系アドレスバス202上のデータが入力される。このデータをもとに、順序回路の内部状態を遷移させると同時に、演算系CPU100の内部アクセス終了信号57を出力する。I/O系内部バス制御回路52には、I/O系アドレスバス204上のデータが入力される。このデータをもとに、順序回路の内部状態を遷移させると同時に、I/O系CPU101の内部アクセス終了信号59を出力する。演算系CPU100の外部アクセス終了信号56と内部アクセス終了信号57は、OR論理ゲート54で論理和がとられ、バスサイクルが完了したことを指令する制御信号24として出力される。同様に、I/O系CPU101の外部アクセス終了信号58と内部アクセス終了信号59は、OR論理ゲート55で論理和がとられ、バスサイクルが完了したことを指令する制御信号25として出力される。演算系CPU100およびI/O系CPU101がデータアクセスをするために出力するアドレスは、図3、図4、図16、図17に示したように、チップ仕様としてそのアドレス・マップが決められている。そのため、そのアドレス・マップによって外部アクセスと内部アクセスの区別が行なえるので、順序回路で構成された3つのバス制御回路50、51、52は制御すべきバスサイクルを認知することができる。例えば、演算系CPU100が外部アクセスを行なっている期間は、演算系内部バス制御回路51は内部データ・アクセス制御を行っていないので、内部アクセス終了信号57は'0'レベルを出力する。外部バス制御回路50に入力されるアクセス権設定レジスタ53のデータは、演算系CPU100とI/O系CPU101とのいずれが外部アクセスの優先権を有するのかを指定する。高速ページ・アクセスが可能なダイナミックメモリ(DRAM)上のアドレスが連続する空間に格納されたデータを大量に読み込む場合には、同一ページ内のデータアクセスが高速化されるため、連続読み出しをした方が、実際に必要となるメモリアクセス時間の総和が少なくなる。また、データ転送サイクルが規定されているデバイスに対するア

クセスでは、バースト・アクセスをしなければならない場合もある。そのため、演算系とI/O系のどちらかのCPUに外部アクセスを専有させるモードを持つことが必要となる。この指定を、I/O系CPU101がI/O系内部データバス201を介してアクセス権設定レジスタ53にモード情報を書き込むことにより行なう。

【0036】図21には、アクセス権設定レジスタ53に、演算系とI/O系の2つのCPUの外部アクセス優先権を等しく設定した場合のタイミング・チャートを示す。T1の期間には、I/O系CPU101と演算系CPU100とは、同時に内部アクセス1、内部アクセスAを行なっている。T2の期間では、I/O系CPU101が外部アクセス2を実行し、同時に演算系CPU100が内部アクセスB、Cを行なっている。T3の期間には、I/O系CPU101の外部アクセス2が継続中なので、演算系CPU100の外部アクセスDが待ち状態となっている。T4の期間では、待ち状態になっていた演算系CPU100の外部アクセスDが実行され、I/O系CPU101の外部アクセス3が待ち状態になっている。この待ち状態になった外部アクセス3は、T5の期間から実行される。このように、このモードでは、2つの演算系CPU100と101の外部アクセスの割合は、均等に設定されている。

【0037】図23には、アクセス権設定レジスタ53で、I/O系CPU101に外部アクセスを専有させるモード(I/O系CPU101への独占的連続使用権)を設定した場合のタイミング・チャートを示す。T1の期間は、I/O系CPU101と演算系CPU100とは、同時に内部アクセス1、内部アクセスAを行なっている。T2の期間では、I/O系CPU101が外部アクセス2を実行し、同時に演算系CPU100が内部アクセスB、Cを行なっている。T8の期間に行なわれようとしている演算系CPU100の外部アクセスDは、アクセス権設定レジスタ53がI/O系CPU101の外部アクセス専有モードの指定を解除するまで、実行されない。そのため、このモードでは、I/O系CPU101が外部アクセスを専有することになる。この例とは逆に、アクセス権設定レジスタ53演算系CPU100に外部アクセスを専有させるモードを作ることが可能であることは言うまでもない。

【0038】次に、図1に示した割込み制御回路17の一実施例として、図24にマルチCPUの割込み系の構成図を示す。図24は、演算系CPU100、I/O系CPU101、割込み制御回路120、割込みフラグレジスタ121、割込み制御回路122、割込みフラグレジスタ123、A/DおよびD/A変換器124、タイマTMR125、シリアル入出力インターフェースSC1126、割込みベクタレジスタ127、128より構成されている。この構成により、I/O系CPU101から演算系CPU100への割込み、および演算系CP

U100からI/O系CPU101への割込みが可能となる。すなわち、演算系内部データバス200は、割込みフラグレジスタ123および割込みベクタレジスタ128に接続されている。割込みベクタレジスタ128は、さらにI/O系内部データバス201に接続されている。割込みフラグ123は、信号線248を通して割込み制御回路122に接続されている。割込み制御回路122には、外部割込み要求信号249および内部割込み要求信号250が接続されている。割込み制御回路122は、割込みフラグクリア信号257を通して割込みフラグ123に接続されている。割込み制御回路122は、割込みベクタ読み出し信号256を通して割込みベクタレジスタ127に接続されている。割込み制御回路122は、割込み要求信号247および割込みベクタ信号258を通してI/O系CPU101に接続されている。I/O系内部データバス201は、割込みフラグレジスタ121および割込みベクタレジスタ127に接続されている。I/O系内部データバス201は、信号線251を通してA/DおよびD/A変換器124に接続され、信号線252を通してTMR125に接続され、信号線253を通してSC1126に接続されている。割込みベクタレジスタ127は、演算系内部データバス200に接続されている。割込みフラグ121は、信号線246を通して割込み制御回路120に接続されている。割込み制御回路120は、割込み要求信号245を通して演算系CPU100に接続されている。割込み制御回路120は、割込みフラグクリア信号254を通して割込みフラグ121に接続されている。割込み制御回路120は、割込みベクタ読み出し信号255を通して割込みベクタレジスタ127に接続されている。以下に動作の説明を行う。演算系CPU100は、演算系内部データバス200を通してベクタアドレスを割込みベクタ128にセットし、さらに、演算系内部データバス200を通して割込みフラグ123に値をセットする。割込みフラグ123は、信号線248を通して割込み制御回路122に伝達される。割込み制御回路122は、信号線248より信号を受けると割込み要求信号247に信号を出力する。同時に割込みベクタ読み出し信号256に信号を出力する。割込みベクタレジスタ128は、割込みベクタ読み出し信号256から信号を受けると、I/O系内部データバス201にベクタアドレスを出力する。I/O系CPU100は割込み要求信号247より信号を受けると、演算系CPU100からの割込みであることを認識し、I/O系内部データバス201に出力されているベクタアドレスを読み込み、この値に従って割込み要求処理を開始する。このとき割込み制御回路122は、I/O系CPU100が割込み要求信号247の信号を受けとったことを認識すると、信号線257に信号を出力して、割込みフラグ123にセットされた値をリセットする。外部割込み要求信号249や内部割

込み要求信号250から割込みが発生した場合、割込み制御回路122は割込み要求信号247に信号を出力すると同時にベクタアドレスを割込みベクタ信号258に出力する。さらに割込みベクタレジスタ128からベクタアドレスが出力されないように、割込み制御回路122は割込みベクタ読み出し信号256に信号を出力しない。I/O系CPU100は割込み要求信号247からの信号を受けると、外部割込みまたは内部割込みであることを認識し、割込みベクタ信号258に出力されているベクタアドレスを読み込み、この値に従って割込み処理を実行する。I/O系CPU100から演算系CPU100へ割込みを発生させる時には、割込み制御回路120、割込みフラグレジスタ123、および割込みベクタレジスタ127により上記と同じ方法で実現することができる。

【0039】図15、図24に示したシングルチップマイコンコンピュータの応用として、音声処理を考える。演算系CPU100が処理すべき入力データの処理手順の様子を、演算系メモリマップの形で図25に示す。図25ではシェアドメモリC115、シェアドメモリD116、シェアドメモリE117の順番にデータが処理される。図24(1)の状態は、現在演算系CPU100が扱っているシェアドメモリを表している。この場合、内部メモリ空間AにはシェアドメモリC115が、内部メモリ空間BにはシェアドメモリD116が接続されている。シェアドメモリC115およびシェアドメモリD116のデータ演算処理が順次終了すると、図24(2)のようにシェアドメモリを接続しなければならない。図24(2)を見ると、(1)の内部メモリ空間AにあったシェアドメモリC115が接続されなくなり、代わりに内部メモリ空間Bに接続されていたシェアドメモリD116が内部メモリ空間Aに接続されている、さらに内部メモリ空間BにはシェアドメモリE117が新たに接続されていることがわかる。音声処理では、データがある大きさと区切って処理する場合、区切った境界部分のデータ演算処理の質を低下させないために、直前に処理したデータを残して、その部分から再度処理を開始する必要がある。そのため図25(1)の処理が終了した後、直前に処理したデータであるシェアドメモリD116のデータを図25(2)の状態ですべてに処理できるようにし、その後新たなデータを記憶しているシェアドメモリE117を接続している。以上の動作を実現するために、以下に図15に示したシングルチップマイコンコンピュータの動作の説明を行う。演算系CPU100にすでに接続されているシェアドメモリにデータの書き込みを行わないために、I/O系CPU101は処理すべきデータをシェアドメモリに入力する前にバンク切換えレジスタ119を読み込み、現在演算系CPU100にどのシェアドメモリが接続されているかを調べる。今は、最初の音声処理データを転送する段階で、演

算系CPU100には、いずれのシェアドメモリも接続されていないものとする。I/O系CPU101は、図21の内部/外部バス制御回路16内のアクセス権レジスタ53に値を設定して、I/O系CPU101が外部アクセスを享有するモードにする。その結果、アクセス権レジスタ53は設定された値を、外部バス制御回路50に出力する。図21の外部バス制御回路50は、アクセス権レジスタ53に設定された値を受けると、制御信号20に信号を出力する。制御信号20の信号は、外部データバス21と外部アドレスバス22がそれぞれI/O系内部データバス201とI/O系アドレスバス204に接続されるようにマルチプレクサ18および19を制御する。I/O系CPU101は外部アクセスを行うために、外部空間のアドレスをI/O系アドレスバス204に出力する。I/O系CPU101からI/O系アドレスバス204に出力されたアドレスは、マルチプレクサ19を通して外部アドレスバス22に出力される。出力されたアドレスによりアクセスされた外部メモリのデータは、外部データバス21、マルチプレクサ18を通してI/O系内部データバス201に入力される。I/O系CPU101は、I/O系内部データバス201に入力されたデータをシェアドメモリに転送する。現在の状態は音声処理に初期であり、演算系CPU100に接続されているシェアドメモリは存在しないので、シェアドメモリC115、シェアドメモリD116の順番に外部から取り込んだデータを転送する。2つのシェアドメモリへのデータ転送が終了したら、I/O系CPU101はデータ転送が終了したシェアドメモリを表すために、図15のバンク切換えレジスタ119に値をセットする。この場合は、シェアドメモリC115およびシェアドメモリD116の順番にデータを格納したので、バンク切換えレジスタ119へは(SA1, SA0, SB1, SB0) = (0, 0, 0, 1)をセットする。次にI/O系CPU101は、シェアドメモリに転送したデータを演算系CPU100が取り込み、データ演算処理することを知らせるベクタアドレスを図24の割込みベクタレジスタ127にセットする。最後にI/O系CPU101は、割込みフラグ121に値をセットする。その後I/O系CPU101は、現在演算系CPU100に接続されていないシェアドメモリE117、シェアドメモリF118の順番に外部データを転送する。一方、図24の割込みフラグレジスタ121に値がセットされると、割込みフラグレジスタ121は信号線246に信号を出力する。割込み制御回路120は信号線246より信号を受けると、割込み要求信号245に割込み要求信号を出力する。同時に割込み制御回路120は、割込みベクタレジスタ127に格納されているベクタアドレスを演算系内部データバス200へ出力するために、ベクタアドレス読み出し信号255に"1"を出力する。割込みベクタレジスタ127はベクタアドレス読み出し

信号255より信号を受け取ると、演算系内部データバス200へベクタアドレスを出力する。演算系CPU100は、割込み要求信号245より割込み要求信号を受けるとそれまでのデータ演算処理を中断し、I/O系CPU101からの割込みであることを認識し、演算系内部データバス200に出力されているベクタアドレスを取り込む。割込み制御回路120は、演算系CPU100が割込みを受け付けたところで、割込みフラグ121にセットされた値をクリアするために割込みフラグクリア信号254に"1"を出力する。割込みフラグレジスタ121は、割込みフラグクリア信号254より信号を受け取ると、割込みフラグをクリアする。演算系CPU100は、取り込んだベクタアドレスより、シェアドメモリに格納されたデータを処理するプログラムを読み込み、プログラムを実行する。この場合、演算系CPU100は内部メモリ空間Aおよび内部メモリ空間Bへアクセスを行う。内部メモリ空間Aへアクセスするとき、演算系アドレスバスのA31とA28は(A31, A28) = (1, 0)となる。このとき(SA1, SA0) = (0, 0)となっているので、シェアドメモリC115を指定するCS0信号224のみに"1"が出力される。従って演算系CPU100は、I/O系CPU101がシェアドメモリC115へ転送したデータを取り込むことができる。さらに演算系CPU100が内部メモリ空間Bへアクセスするときは、演算系アドレスバスのA31とA28は(A31, A28) = (1, 1)となる。このとき(SA1, SA0) = (0, 1)となっているので、シェアドメモリD116を指定するCS1信号225のみに"1"が出力される。従って演算系CPU100は、I/O系CPU101がシェアドメモリD116へ転送したデータを取り込むことができる。以上のことより、I/O系CPU101がシェアドメモリC115およびシェアドメモリD116へ転送したデータを演算系CPU100へ引き渡すことができ、図25(1)の状態を実現することができる。演算系CPU100は、I/O系CPU101が外部データを取り込み、メモリに格納して割込み要求信号を発生するまで本来のデータ演算処理を実行できるので、データ演算処理効率の低下を防ぐことが可能になる。演算系CPU100は、シェアドメモリC115およびシェアドメモリD116のデータを取り込み終わったら、演算系内部データバス200を通して割込みベクタレジスタ128にベクタアドレスをセットする。次に、演算系CPU100は、演算系内部データバス200を通して割込みフラグレジスタ123に値をセットした後、演算系CPU100は本来のデータ演算処理を続行する。割込みフラグレジスタ123は信号をセットされると、信号線248にセットされた信号を出力する。割込み制御回路122は、信号線248から信号を受けると演算系CPU100からの割込みであることを認識し、割込み要求信号2

47に割込み要求信号を出力する。同時に割込み制御回路122は、割込みベクタ読み出し信号256に信号を出力する。割込みベクタ読み出し信号256から信号を受けた割込みベクタレジスタ128は、I/O系内部データバス201に値を出力する。このとき割込み制御回路122は、割込みベクタ信号258にベクタアドレスを出力しない。I/O系CPU101は、割込み要求信号247より割込み要求信号を受けると、演算系CPU100からの割込みであることを認識し、I/O系内部データバス201に出力されているベクタアドレスを取り込む。割込み制御回路122は、I/O系CPU101が割込みを受け付けたところで、割込みフラグクリア信号257に信号を出力して、割込みフラグレジスタ123をクリアする。I/O系CPU101は、取り込んだベクタアドレスより演算系CPU100がシェアドメモリのデータを取り込み終わった後の処理プログラムを実行する。I/O系CPU101がシェアドメモリE117へのデータ転送がまだ終了していない場合には、データ転送を続けて行う。一方、シェアドメモリE117へのデータ転送が終了している場合には、図25(2)の状態にするために以下のような動作を行う。I/O系CPU101は、バンク切換えレジスタ119へ新たに値をセットする。この場合は、(SA1, SA0, SB1, SB0) = (0, 1, 1, 0)をセットし、内部メモリ空間AにはシェアドメモリD、内部メモリ空間BにはシェアドメモリEを割り当てる。音声処理では、サンプルデータを適当な長さに区切って処理した場合、次のデータの処理を始める前に直前に処理したデータの処理をもう一度行ってから、新たなシェアドメモリのデータの処理に移らなければならない。これは、データを区切った境界部分での音声処理の質を低下させないためである。そのためにシェアドメモリDを内部メモリ空間Aへ移動させた。バンク切換えレジスタ119への値のセットが完了したら、I/O系CPU101は上記と同じ手順で演算系CPU100へ割込み要求を出す。演算系CPU100は、I/O系CPU101より割込み要求信号を受けた場合、演算系内部データバス200よりベクタアドレスを取り込む。取り込んだベクタアドレスより、シェアドメモリのデータを取り込むプログラムを入力し、プログラムを実行する。この場合、内部メモリ空間AにはシェアドメモリD116が、内部メモリ空間BにはシェアドメモリE117が接続されている。ここで先程と異なる点は、シェアドメモリD116のデータは、すでに処理が終了しているということである。従ってシェアドメモリD116の最初から処理を行うと同じ処理を二度行うことになる。そこでこの場合、割込み要求により取り込んだプログラムの中で、実際の処理を例えばH'80000000以後のプログラマが必要と思われる任意のアドレスから開始するようにする。ここから処理を始めて、シェアドメモリの値を取り込み終わっ

たら、再び上記と同じ手順により、演算系CPU100からI/O系CPU101へ割込み要求を出す。I/O系CPU101は、演算系CPU100から割込み要求を受けたら、新たにシェアドメモリにデータをセットし、演算系CPU100へ引き渡す。以上の動作を繰り返すことにより、CPUのデータ演算処理効率を低下させることなく、音声処理を実行できる。このように、本応用例では、バンク切換えレジスタ、シェアドメモリ、CPUから読み書き可能な割込みフラグと割込みベクタレジスタを備えたマルチCPUシングルチップマイコンを用いることにより、音声処理の効率を低下させずに実行させることが可能である。さらに本発明の構成を1チップ上に実現することにより、複数のチップで実現する場合に比べ高速動作が可能となる。さらに複数のチップで実現する場合に比べ、1チップのピン数を削減することができる。さらに複数のチップで実現する場合に比べ、安価に供給することができる。尚、本発明では、4つのシェアドメモリを用いて説明を行った。バンク切換えレジスタのビット数をシェアドメモリの数を表わせるだけ設ければ、4つ以上のシェアドメモリについても対応できる。また、アドレスデコーダへ入力するアドレス信号の数を増加させれば、メモリを割り当てる内部メモリ空間の数を増加させることが可能である。またこのときの一つのメモリ空間の大きさを小さくすることも可能である。

【0040】

【発明の効果】本発明によれば、複数のCPUを有するシングルチップマイクロコンピュータにおいて、周辺モジュールからの割込み要求処理を高速に実行するとともに、CPUのデータ演算処理の効率を向上させることができる。

【図面の簡単な説明】

【図1】本発明の実施例によるマルチCPUを示す図である。

【図2】図1のマルチCPUのシェアドメモリ部の示す図である。

【図3】図1の演算系CPUのメモリマップを示す図である。

【図4】図1のI/O系CPUのメモリマップを示す図である。

【図5】図2のメモリ割当てレジスタを示す図である。

【図6】図2のマルチCPUのモード指定とメモリマップとの関係を示す図である。

【図7】演算系CPU側のアドレスデコーダのブロック図である。

【図8】演算系CPU側のアドレスデコーダの機能を示す図である。

【図9】I/O系CPUのアドレスデコーダのブロック図である。

【図10】I/O系CPUのアドレスデコーダの機能を

示す図である。

【図11】選択回路のブロック図である。

【図12】選択回路の機能を示す図である。

【図13】シェアドメモリの構成を示す図である。

【図14】他のメモリ割当てレジスタの構成を示す図である。

【図15】シェアドメモリ部におけるバンク切換えを実現するマルチCPUの実施例を示す図である。

【図16】演算系CPUのメモリマップを示す図である。

【図17】I/O系CPUのメモリマップを示す図である。

【図18】バンク切換えレジスタの構成を示す図である。

【図19】演算系CPU側のアドレスデコーダのブロック図である。

【図20】演算系CPU側のアドレスデコーダの機能を示す図である。

【図21】内部/外部バス制御回路の構成を示す図である。

【図22】アクセス権設定レジスタに2つのCPUの外部アクセスを等しく設定した場合のタイミング・チャートを示す図である。

【図23】アクセス権設定レジスタにI/O系CPUが外部アクセスを専用するモードを設定した場合のタイミング・チャートを示す図である。

【図24】割込み制御回路の構成をより詳細に示す図である。

【図25】音声処理実行時における演算系CPUメモリマップの変化を示す図である。

【符号の説明】

10…内蔵メモリ、11…タイマ、12…A/DおよびD/A変換器、13…シリアル入出力インタフェース、14…乗除算器、15…関数演算器、16…内部/外部バス制御回路、17…割込み制御回路、18…マルチプレクサ、19…マルチプレクサ、20…制御信号、21…外部データバス、22…外部アドレスバス、23…WAIT制御信号、24…制御信号、25…制御信号、50…外部バス制御回路、51…演算系内部バス制御回路、52…I/O系内部バス制御回路、53…アクセス権設定レジスタ、54…OR回路、55…OR回路、56…外部アクセス終了信号、57…内部アクセス終了信号、58…外部アクセス終了信号、59…内部アクセス終了信号、100…演算系CPU、101…I/O系CPU、102…アドレスデコーダ、103…アドレスデコーダ、104…シェアドメモリA、105…シェアドメモリB、107…選択回路、108…メモリ割当てレジスタ、109…メモリ、110…選択回路、111…選択回路、112…選択回路、113…アドレスデコーダ、114…アドレスデコーダ、115…シェアドメモ

29

(16)

特開平7-84963

リC、116…シェアメモリD、117…シェアメモリE、118…シェアメモリF、119…バンク切換えレジスタ、120…割込み制御回路、121…割込みフラグ、122…割込み制御回路、123…割込みフラグ、124…A/D、125…TMR、126…SC1、127…割込みベクタレジスタ、128…割込みベクタレジスタ、129…パッド、130…パッド、131…パストランジスタ、132…パストランジスタ、200…演算系内部データバス、201…I/O系内部データバス、202…演算系アドレスバス、203…演算系上位アドレス、204…I/O系アドレスバス、205…I/O系上位アドレス、206…信号線、207…信号線、208…信号線、209…信号線、210…信号線、211…CS4信号、212…CS5信号、213…CS6信号、214…CS7信号、215…SL0信号、216…SL1信号、217…信号線、218…信号線、219…信号線、220…演算系下位アドレス、221…演算系下位アドレス、222…I/O系下位アドレス、223…I/O系下位アドレス、224…*

10

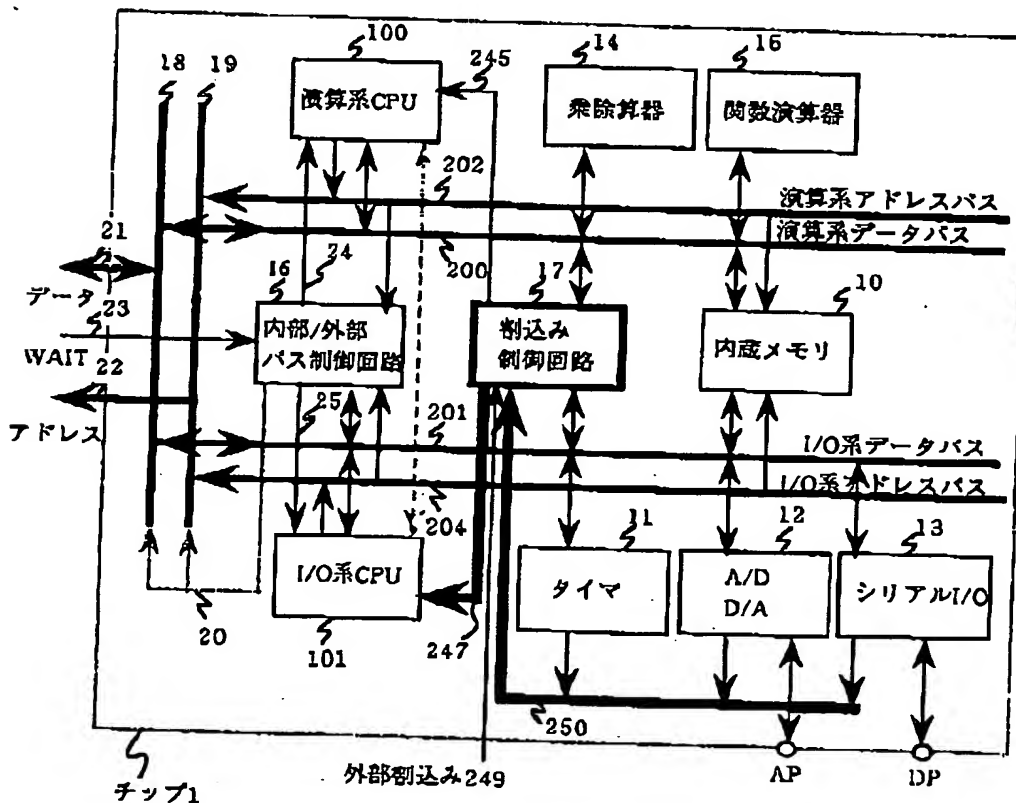
*CS0信号、225…CS1信号、226…CS2信号、227…CS3信号、228…CS8信号、229…CS9信号、230…CS10信号、231…CS11信号、232…演算系下位アドレス、233…演算系下位アドレス、234…演算系下位アドレス、235…演算系下位アドレス、236…I/O系下位アドレス、237…I/O系下位アドレス、238…I/O系下位アドレス、239…I/O系下位アドレス、240…信号線、241…SA1信号、242…SA0信号、243…SB1信号、244…SB0信号、245…割込み要求信号、246…信号線、247…割込み要求信号、248…信号線、249…外部割込み要求信号、250…内部割込み要求信号、251…信号線、252…信号線、253…信号線、254…割込みフラグクリア信号、255…割込みベクタ読み出し信号、256…割込みベクタ読み出し信号、257…割込みフラグクリア信号、258…割込みベクタ読み出し信号、259…書き込み信号線、260…信号線、261…信号線、262…信号線、263…信号線。

【図1】

図1

【図5】

図5

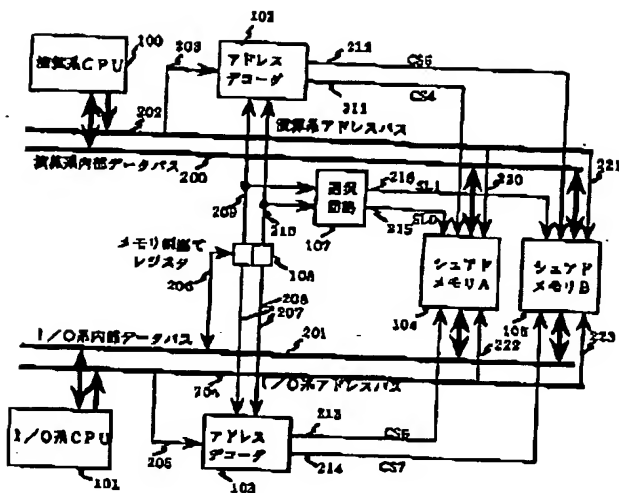


(17)

特開平7-84963

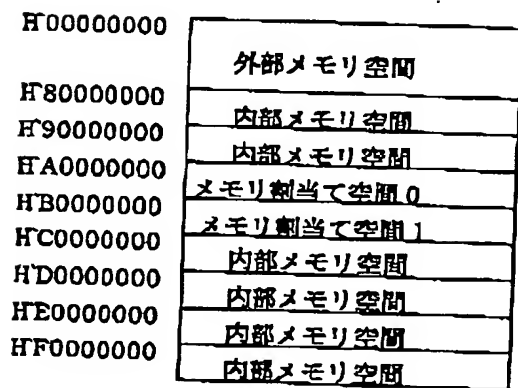
【図2】

図2



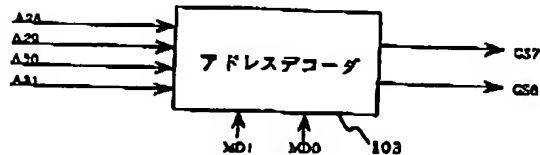
【図4】

図4



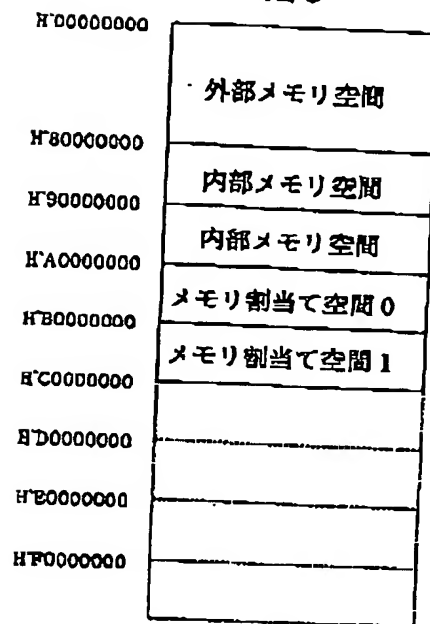
【図9】

図9



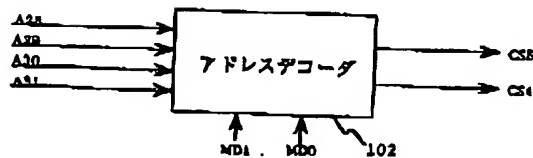
【図3】

図3



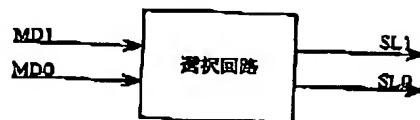
【図7】

図7



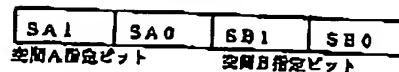
【図11】

図11



【図18】

図18

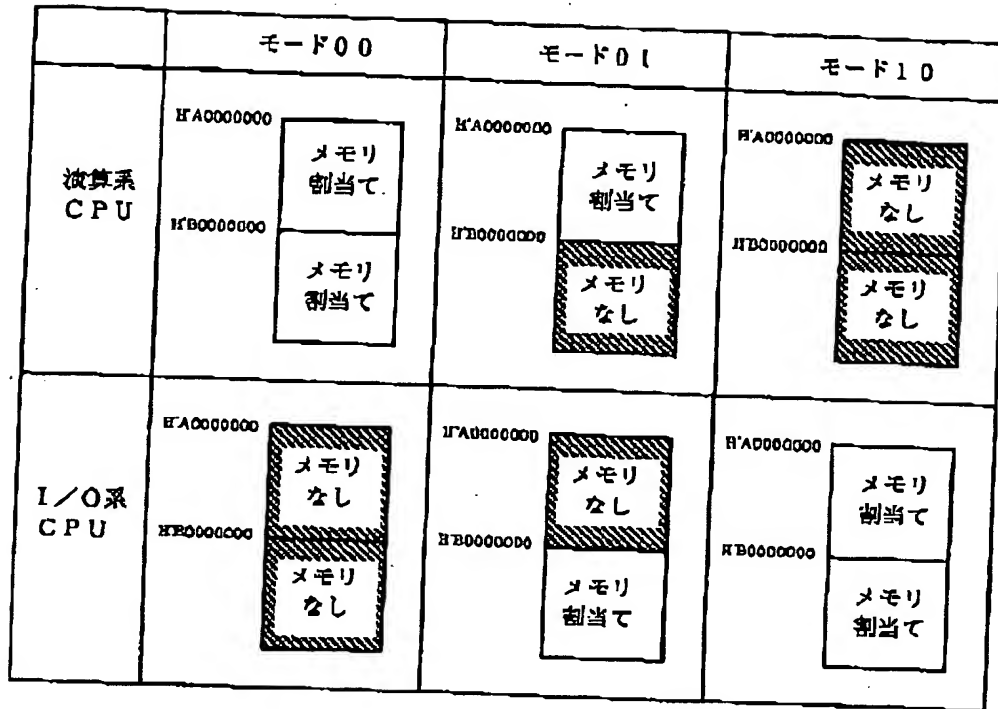


(18)

特開平7-84963

【図6】

図 6



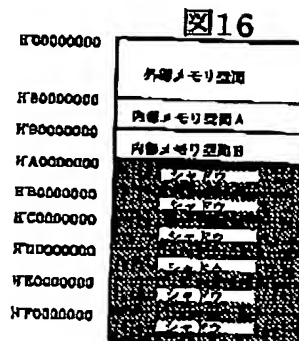
【図12】

図 1 2

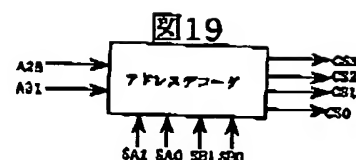
入 力		出 力		機 能
モード指定		(1)	(2)	
MD1	MD0	SL0	SL1	
0	0	1	1	シェアドメモリA、Bを演算系CPUに接続
0	1	1	0	シェアドメモリAのみ演算系CPUに接続
1	0	0	0	演算系CPUにはシェアドメモリを接続しない
1	1	*	*	サポートしていないモード指定

- (1) シェアドメモリAへ入力されている演算系、I/O系のアドレスバスおよびデータバスの選択信号
 (2) シェアドメモリBへ入力されている演算系、I/O系のアドレスバスおよびデータバスの選択信号

【図16】



【図19】



【図8】

図 8

入 力						出 力		機 能
アドレス				モード指定		メモリセレクト		
A31	A30	A29	A28	MD1	MD0	CS4	CS5	
0	*	*	*	*	*	0	0	外装空間指定
1	1	*	*	*	*	0	0	EC00000000-HFFFFFFF空間指定
1	0	0	*	*	*	0	0	内部メモリ空間指定 (H80000000-H9FFFFFFF)
1	0	1	0	0	0	1	0	メモリ割当て空間0、モード00指定
1	0	1	0	0	1	1	0	メモリ割当て空間0、モード01指定
1	0	1	0	1	0	0	0	メモリ割当て空間0、モード10指定
1	0	1	0	1	1	*	*	メモリ割当て空間0、サポートしていないモード指定
1	0	1	1	0	0	0	1	メモリ割当て空間1、モード00指定
1	0	1	1	0	1	0	0	メモリ割当て空間1、モード01指定
1	0	1	1	1	0	0	0	メモリ割当て空間1、モード10指定
1	0	1	1	1	1	*	*	メモリ割当て空間1、サポートしていないモード指定

【図10】

図 1 0

入 力						出 力		機 能
アドレス				モード指定		メモリセレクト		
A31	A30	A29	A28	MD1	MD0	CS4	CS7	
0	*	*	*	*	*	0	0	外装空間指定
1	1	*	*	*	*	0	0	内部メモリ空間指定 (HCC0000000-HFFFFFFF)
1	0	0	*	*	*	0	0	内部メモリ空間指定 (H80000000-H9FFFFFFF)
1	0	1	0	0	0	0	0	メモリ割当て空間0、モード00指定
1	0	1	0	0	1	0	0	メモリ割当て空間0、モード01指定
1	0	1	0	1	0	1	0	メモリ割当て空間0、モード10指定
1	0	1	0	1	1	*	*	メモリ割当て空間0、サポートしていないモード指定
1	0	1	1	0	0	0	0	メモリ割当て空間1、モード00指定
1	0	1	1	0	1	0	1	メモリ割当て空間1、モード01指定
1	0	1	1	1	0	0	1	メモリ割当て空間1、モード10指定
1	0	1	1	1	1	*	*	メモリ割当て空間1、サポートしていないモード指定

【図17】

図17

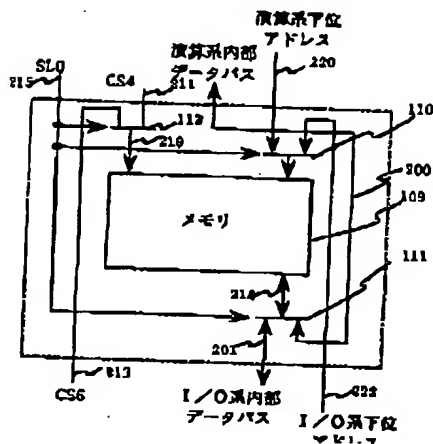
H00000000	外部メモリ空間
H80000000	内部メモリ空間
H90000000	内部メモリ空間
HA00000000	内部メモリ空間
HB00000000	内部メモリ空間
HC00000000	シェアDメモリC
HD00000000	シェアDメモリD
HE00000000	シェアDメモリE
HF00000000	シェアDメモリF

(20)

特開平7-84963

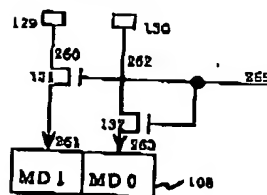
【図13】

図13



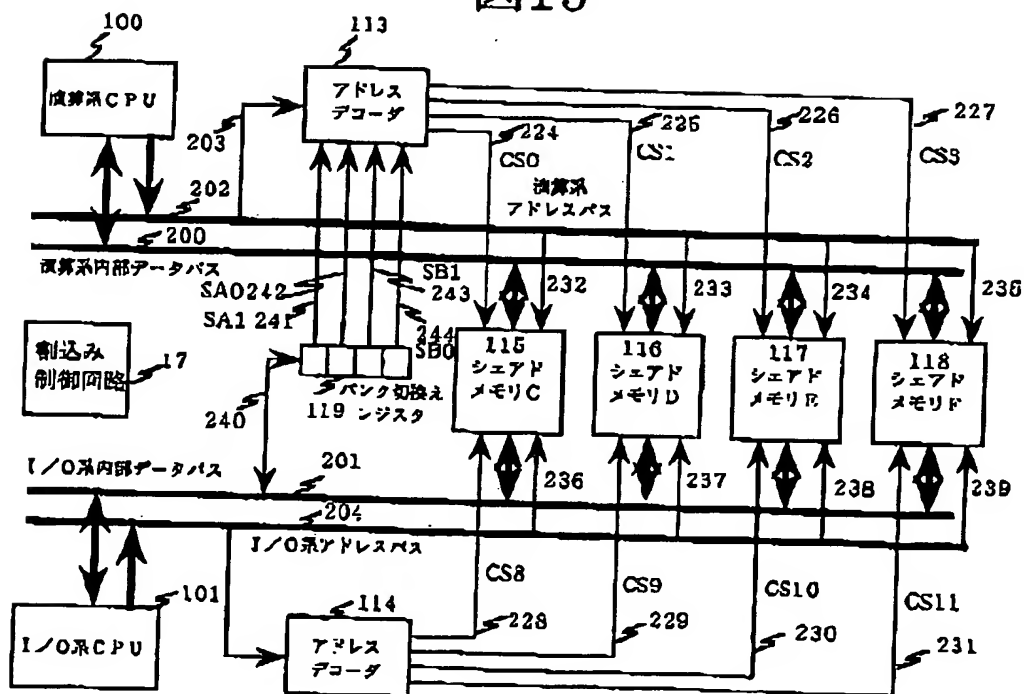
【図14】

図14



【図15】

図15



(21)

特開平7-84963

【図20】

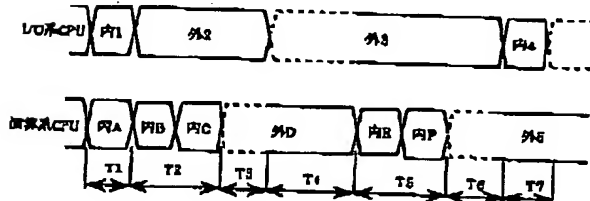
図20

図 20

入 力						出 力				機 能	
アドレス		空間A指定		空間B指定		メモリセレクト					
A31	A28	SA1	SA0	SB1	SB0	CS0	CS1	CS2	CS3		
0	*	*	*	*	*	0	0	0	0	外部空間指定	
1	0	0	0	*	*	1	0	0	0	空間A、シェアドメモリC指定	
1	0	0	1	*	*	0	1	0	0	空間A、シェアドメモリD指定	
1	0	1	0	*	*	0	0	1	0	空間A、シェアドメモリE指定	
1	0	1	1	*	*	0	0	0	1	空間A、シェアドメモリF指定	
1	1	*	*	0	0	1	0	0	0	空間B、シェアドメモリC指定	
1	1	*	*	0	1	0	1	0	0	空間B、シェアドメモリD指定	
1	1	*	*	1	0	0	0	1	0	空間B、シェアドメモリE指定	
1	1	*	*	1	1	0	0	0	1	空間B、シェアドメモリF指定	

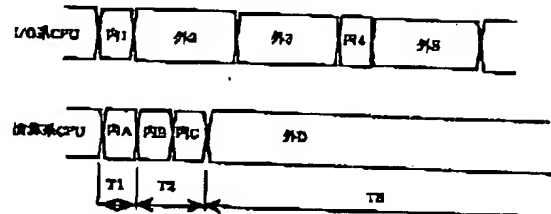
【図22】

図22



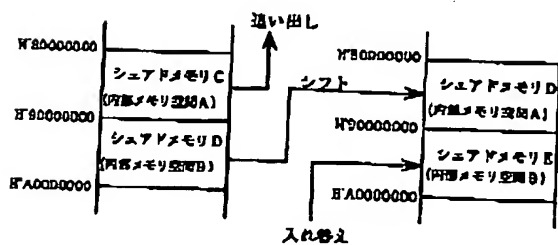
【図23】

図23



【図25】

図25



(1) 入力処理

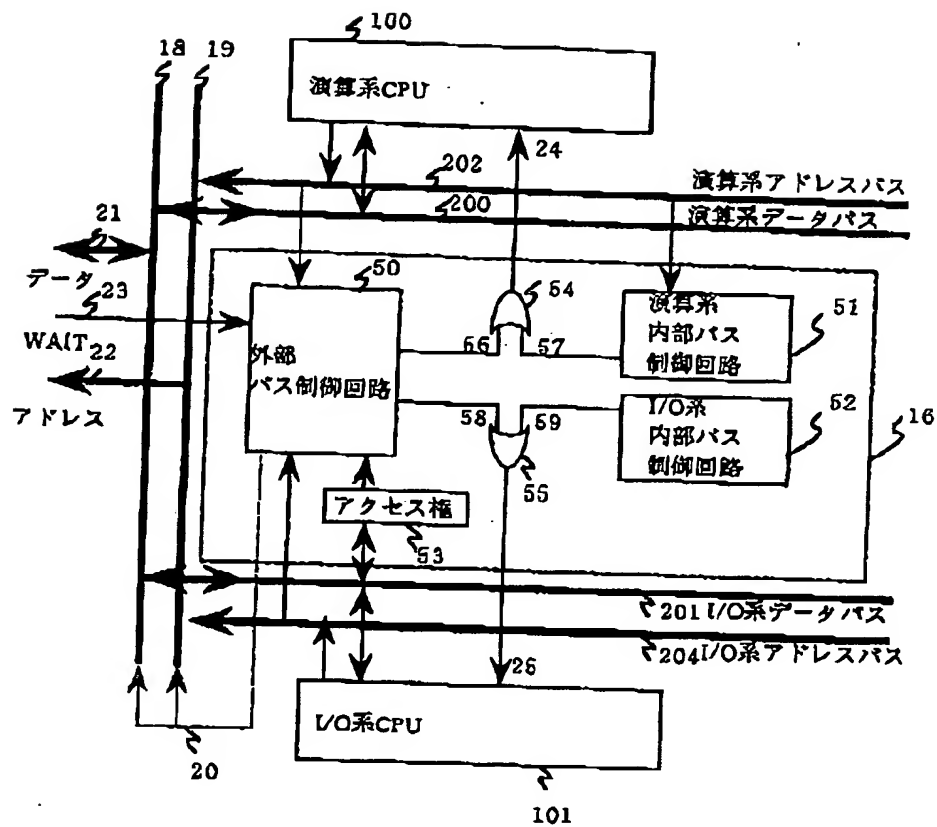
(2) 出力処理

(22)

特開平7-84963

【図21】

図21

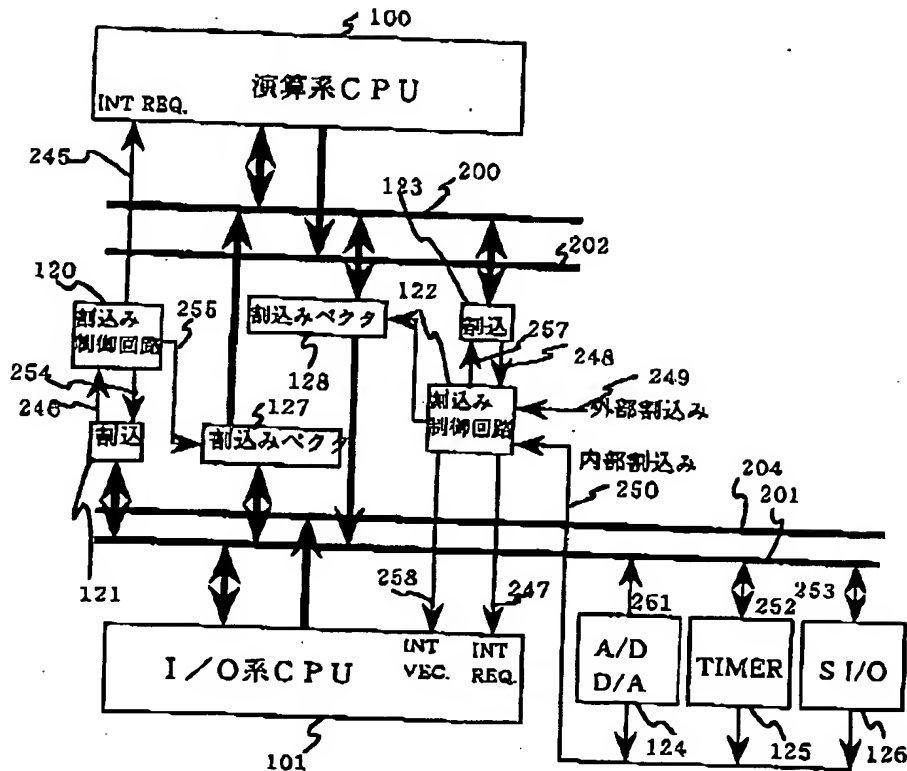


(23)

特開平7-84963

【図24】

図24



フロントページの続き

(72)発明者 棚寝 義人

東京都国分寺市東恋ヶ窪1丁目280番地
株式会社日立製作所中央研究所内

(72)発明者 池田 宏

東京都国分寺市東恋ヶ窪1丁目280番地
株式会社日立製作所中央研究所内

(72)発明者 天野 明雄

東京都国分寺市東恋ヶ窪1丁目280番地
株式会社日立製作所中央研究所内

(72)発明者 上牧 春雄

東京都国分寺市東恋ヶ窪1丁目280番地
株式会社日立製作所中央研究所内

(72)発明者 浅川 吉章

東京都国分寺市東恋ヶ窪1丁目280番地
株式会社日立製作所中央研究所内

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.